SANDIA NATIONAL LABORATORIES
QUALITY ASSURANCE PROGRAM
for the
OFFICE of CIVILIAN RADIOACTIVE WASTE MANAGEMENT

QAP 19-1

SOFTWARE DEVELOPMENT and USE

Revision 0

Effective Date: 06/08/2004

Author:	*Original Signed by Martha J. Mitchell*	Date:	*06/08/2004*
Concurrence:	*Original Signed by Jose A. Archuleta* QA Reviewer	Date:	*06/08/2004*
Approval:	*Original Signed by M. Kathryn Knowles* SNL OSTI Project Lead	Date:	*06/08/2004*

CHANGE HISTORY

Revision	Description	Effective Date
0	This is the initial version of this document.	06/08/2004

TABLE OF CONTENTS

Page

1.0 Purpose and Scope

This procedure prescribes the processes used to qualify and control software used in the Sandia National Laboratories (SNL) Office of Science and Technology and International (OSTI) program. This procedure for software development and use follows the approach established by the Capability Maturity Model (CMM) and conventional Software quality assurance including life-cycle management. It is expected that this procedure will under go revision as the software development process improves and becomes more mature.

This procedure provides the process for allowing software to be used in applications governed by the SNL OSTI QA Program (Q), and covers the following areas. This list identifies the reference in this and other procedures where the requirements and processes for the activity are described.

- Request for use of qualified software from the YMP qualified baseline. Complete form LSI11-2, Software User Request and follow procedure LP-SI-11Q-BSC for installation and checkout.
- Qualification and use of acquired software off-the-shelf and developed by others
- Development and qualification of new software applications including numerical and software models
- Development and use of routine calculations and applications
- Software that is used for in managing information
- Software used to collect data for experimental activities or
- Process data from experiments or other OSTI plans or procedures. (Data Acquisition System (DAS) software must be qualified. If the DAS software is an integral part of an off-the-shelf system and not modified, refer to QAP 20-1 (Test Plans under data quality control) for its qualification requirements. If the DAS software is developed or modified for use in the SNL OSTI program, the qualification process described in the body of this procedure must be followed).

Software models range from conceptual to numerical software. These are developed in the same process as other applications of the same function. Conceptual models are reviewed as documents. Existing numerical models are developed reviewed as routine calculations or as software under development. The progress of the model from the conceptual model, to the mathematical model, to an abstraction model, to a systems model shall be made in the documentation to assure traceability of the process and the inclusion of the identified requirements.

This procedure does not apply to software use that is not under the SNL OSTI QA program (i.e. non-Q). However, if an activity or data involving software use is to be used for Q applications then the software shall be fully qualified. To accomplish this, a revision history shall be maintained or the code version with inputs and output or scientific or engineering software shall be included in the code documentation records.

Software governed by this procedure shall comply with the applicable requirements of this procedure prior to use.

Exempt from this procedure are:

- Commercial-off-the-shelf (COTS) System software such as operating systems, administrative and management systems (database management systems), system utilities, assemblers, compilers, interpreters, etc.

- COTS application software such as Microsoft Office, graphics applications, application utilities. (Computer-aided software engineering (CASE) and development tools shall be qualified and included into this procedure).
- Software written to conduct routine calculations and other limited applications which can be verified by hand calculations. Use and qualification of these programs is discussed in the analysis portions procedure QAP 20-1. The analysis may be documented in a scientific notebook, QAP 20-2.

Acronyms and definitions for terms used in this procedure may be found in the OSTI Glossary.

2.0 Implementation Actions

This section contains step-by-step processes for the acquisition, development, maintenance, configuration management, and software problem reporting for software. General requirements that apply to the sub-sections are listed below. The user of this procedure should read and understand these steps prior to implementation of any of the sub-sections.

2.1 General Requirements

1. Quality requirements and documentation for a generic software lifecycle are summarized in Table 1.
2. The review processes may cause portions of the current phase and/ or previous phases to be modified. In such cases, changes to baseline documents shall be made and verified at the same level of detail as the original document(s).
3. All QA records produced by this procedure are assigned a version identifier composed of three parts as needed, each separated by a period. This system is described below:

   Version **X.Y**.**Z** X is the major field. Y and Z are the minor field(s), where Z is used for patches. X, Y, and Z can be alphanumeric characters of any length, e.g., 2.3.8, 1.01.C, 12.2B. These version identifiers are changed when new releases of software and/or baseline documents are released. Baseline documents (e.g. RD, VVP, DD, ID, UM, and VD) with the same major version identifiers shall be consistent with each other, however, the major version identifier of the code need not be the same as the major version identifier of the baseline documents.

2.2 Software Life-Cycle Classification and General Documentation Requirements

The life-cycle phases described in this procedure are:

- Planning,
- Requirements,
- Design,
- Implementation,
- Validation,
- Installation and Checkout,
- Maintenance, and
- Retirement.

The activities associated with the evolution of the software shall use an iterative or sequential approach.

Note: Each phase follows the process flowchart in Appendix A.

There are two classifications of software that follow life-cycle methodology phases, Acquired and Developed. The table below lists applicable requirements for each of these two types of software. Figure 1 shows the documentation flow.

Table 1. Software Requirements

PHASE	Planning	Requirements		Design	Implementation		Validation	I&C			Maintenance		Retirement
LOCATION	2.3.1	2.3.2		2.3.3	2.3.4		2.3.5	2.3.6			2.3.7[5]		2.3.8
APPLICABLE DOCUMENT	SQAP	RD[3]	VVP	DD[4]	ID	UM[2]	VD	I & C	AC	AU	CC[a]	SPR	
FORMS[1] **QAP 19-1-X**	1	2	3	4	5	6	7	8	-	-	9	10	-
Acquired	X	X	X	-	[b]	X	X	X	X	X	X	X	X
Developed	X	X	X	X	X	X	X	X	X	X	X	X	X

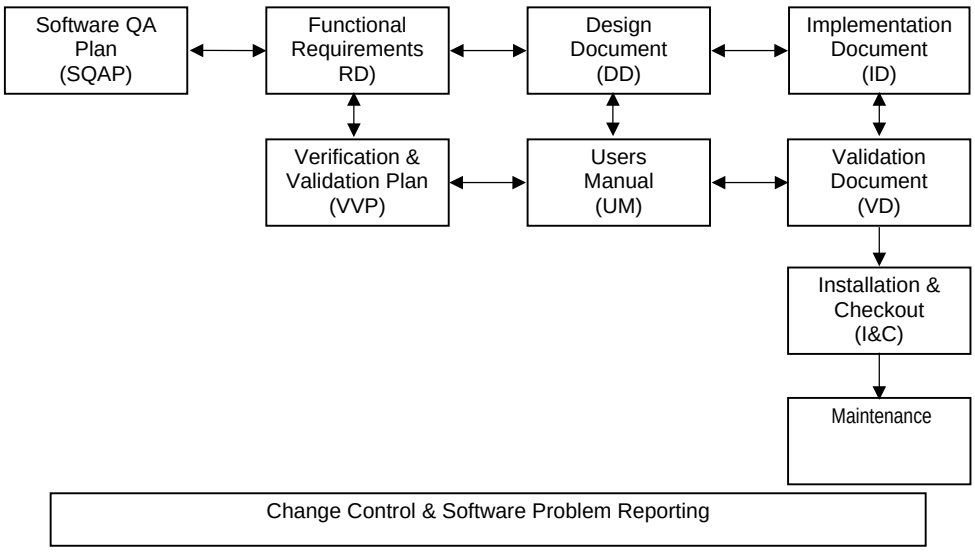
KEY: - indicates that the item is NOT required.
X indicates that the item IS required.
a- the CC and SPR are forms only, not documents
AC refers to Access Control Memorandum
AU refers to Approved Users Memorandum
b- Not applicable when the source code is not acquired

TABLE NOTES

1. All form numbers on the table are preceded by QAP 19-1-X.
2. User Manual (UM) requirements may be fulfilled by referencing and using supplied user instruction publications as long as the supplied documentation complies with the requirements of this procedure.
3. If the requirements of a particular baseline document are provided in multiple documents, a clear path to the fulfillment of the requirements shall be provided.
4. A Design Document (DD) is not initially required for acquired software. If Acquired software is to be modified and the change is significant, a High Level "as built" Design may be developed for the entire existing system depending on the licensing and contract agreements. If the modification is not significant in nature, then a detailed design document is required only for the new portions of the design.
5. Change Control (CC) and Software Problem Reporting (SPR) are required as needed (i.e., when changes to baseline documents are needed or when bugs are discovered.)

Figure 1

Documentation Development Flow



2.2.1 Acquired Software

Commercial Off the Shelf Software (Excel, Mathematica, Word etc.,) as received is acquired software and is exempt from the software development requirements. Numerical Modeling software may be acquired software but is not exempt from the software development requirements. The acquisition of Contracted Software shall follow QAP 4-1 “Procurement”. The procurement requirements shall identify needed documentation developed by the supplier. When qualified software from the YMP qualified baseline is to be used. Form LSI11-2, Software User Request is to be completed and procedure LP-SI-11Q-BSC shall be followed for installation and checkout. If special purpose or modeling software is Commercial and has a Sandia site license it shall be installed from that source and included in the SNL OSTI Qualified Software Baseline List. If no license is available, a new copy shall be procured and the software included on the qualified list for tracking.

Prior to use, Acquired Software shall be evaluated against the life cycle phases. The process of qualifying such software for use is provided below.

The Code Team/Sponsor shall review the adequacy of the primitive baseline to determine the following in accordance with the QA measures identified in Table 1 for Acquired Software:

A. Adequacy of existing verification and validation and software documentation to support operation and maintenance is defined by criteria found in Sections 2.3.1 through 2.3.8.

B. Activities to be performed and the documentation necessary to accept the software for its intended use and place it under configuration control is defined by criteria found in Sections 2.3.1 through 2.3.8.

The following documents are produced following the process described in Appendix A:

- The Primitive Baseline, which consists of the Transition Memo and existing software documentation.
- The transition memo is written by the Code Team/Sponsor to compare existing documentation with requirements.

Following the process described in Appendix A, the Code Team/Sponsor shall submit the primitive baseline to the technical reviewer.

The technical reviewer shall review the primitive baseline to ensure that all applicable requirements are satisfied, or planned to be satisfied. Upon resolution of comments, the technical reviewer signs the Transition Memo.

The Code Team/Sponsor shall forward the reviewed and approved primitive baseline to the SCM Coordinator. The Code Team/Sponsor shall take action to meet applicable QA requirements identified in the Transition Memo.

The SCM Coordinator shall forward the approved primitive baseline to the SNL OSTI Records Center, and redline/update the Software Baseline List as needed.

2.3 Description of Procedure for Each Lifecycle Phase

2.3.1 Planning Phase

A Software QA Plan (SQAP) is produced during this phase for new software development. Software under configuration control and developed within the scope of these QA requirements will not require a stand alone SQAP.

The SQAP shall identify:

- The software to which it applies, objectives of the software, Use cases or problem statements, necessary for the development action. The documents to be prepared, reviewed and maintained during the software life cycle, and their relationship to QA measures defined in this procedure.
- If any deviations from the documentation required by QAP19-1 are anticipated, e.g., a database may not need an Implementation and Validation Document; the SQAP should contain a detailed explanation of how the intent of lifecycle reporting will be met. For efficiency, documents may be merged into combined reports.
- The organizations and/or individuals responsible for performing the work and achieving software quality and their tasks with a schedule for qualification and responsibilities.
- The standards, conventions, techniques, or methodologies that guide software development, as well as the methods used to assure implementation of requirements
- The procedure(s) (QAPs, EIPs, etc.) used for establishing and maintaining the integrity of data, embodied mathematical models, and output files
- The process for reporting and documenting software discrepancies, evaluating the impact of errors on previous calculations, and determining the appropriate corrective action.

Following the development of the SQAP, no strict sequence of performing activities is required provided that all specified requirements for each phase are met and the intent of the requirements are not subverted.

SQAP may be written for an individual code or a set of codes. It should be developed by Code Team/Sponsor and approved (by signature) by the Responsible Manager, Technical Reviewer(s), and the SCM Coordinator following the process described in Appendix A, using the phase criteria listed on the Software QA Plan Criteria Form QAP 19-1-1, (Appendix B).

2.3.2 Requirements Phase

The following documents are produced during this phase:

- Requirements Document (RD) – defines the requirements that the proposed software must satisfy, and
- Verification and Validation Plan (VVP) - identifies tests to be performed and associated acceptance criteria to ensure verification of each software development phase and validation of the entire software baseline.

The Code Team/Sponsor shall develop the RD and VVP following the process described in Appendix A, using the phase criteria listed on the Requirements Document Criteria Form QAP 19-1-2, (Appendix C) and Verification and Validation Plan Criteria Form QAP 19-1-3, (Appendix D). The RD shall be approved prior to the approval of the VVP.

2.3.3 Design Phase

The Design document (DD), produced during this phase, provides the following information (as applicable):

- Theoretical basis (physical process represented),
- Mathematical model (numerical model),
- Control flow and logic,
- Data structures,
- Functionalities and interfaces of objects, components, functions, and subroutines,
- Ranges for data inputs and outputs, in a manner that can be implemented into software.

The Code Team/Sponsor shall develop the DD following the process described in Appendix A, using the phase criteria listed on the Design Document Criteria Form QAP 19-1-4, (Appendix E). The design may necessitate the modification of the RD and VVP.

Note: There may be more than one design document (which may be combined into one document) created during software development. For example a high-level design may be developed to match the code design to the requirements, and define the overall architecture of the code (define modules and subroutines and their purpose, define data structures, define what routine calls what routine, etc.). Another detailed design document may be developed to define how the modules will function in detail, define call interfaces between routines, defines data types, etc. A detailed design as its name implies, is very detailed down to level of almost writing the code (pseudocode).

2.3.4 Implementation Phase

The following documents are produced during this phase:

- The Implementation Document (ID) - provides the source code listing and the process of generating executable software, and
- The User's Manual (UM) - provides information to assist users understanding and using the software.

The design as described in the DD is used as the basis for the software development, and may need to be modified to reflect changes identified in the implementation phase.

The Code Team/Sponsor shall develop the ID and UM following the process described in Appendix A using the phase criteria listed on the Implementation Document Criteria Form QAP 19-1-5, (Appendix F) and the User Manual Criteria Form QAP 19-1-6 (Appendix G).

2.3.5 Verification and Validation Phase

The verification and validation activity shall be independent of development. When this independence cannot be achieved, the Responsible Manager shall document this condition and provide a justification of the individual assigned the test function.

Verification shall be performed on project modules and on changes. Verification shall be performed at the end of the lifecycle phases (requirements, design, implementation, and testing) to ensure traceability and the inclusion of requirements.

Validation Document (VD), produced during this phase, documents the test case input and output files, and the evaluation of the results versus the acceptance criteria identified in the approved VVP for each test case.

The validation phase consists of executing and reviewing the test cases identified in the approved VVP to demonstrate that the developed software meets the requirements defined for it in the RD. The Code Team/Sponsor shall develop and approve the Validation Document following the process described in Appendix A, using the phase criteria listed on the VD Document Criteria Form QAP 19-1-7, (Appendix H).

2.3.6 Installation and Checkout Phase

The following documents are produced during this phase:

- The Installation and Checkout (I&C) Form QAP 19-1-8 (Section 2.3.6.1 and Appendix I)
- The Access Control Memorandum (Section 2.3.6.2) and
- The Approved Users Memorandum (Section 2.3.6.3).

2.3.6.1 The Installation and Checkout Form

The I&C Form provides evidence of:

- The execution of the validation cases on the production computer,
- The installation of the baseline software on the production computer (re-compiling and linking if necessary), and
- The performance of testing with selected test cases (those identified as appropriate for installation and checkout) from the approved VVP to demonstrate acceptable performance on the target computer.

The Code Team/Sponsor shall produce the I&C Form, QAP 19-1-8 (Appendix I) following the process described in Appendix A.

Note: When software is installed (Section 2.4), an Installation and Checkout Form and the Implementation Document are submitted to the Software Configuration Management Coordinator by the Code Team/Sponsor. Completion of required software QA documentation may require critical code modifications.

Note: Installation on a network of identical computers running identical operating systems requires testing on only one of the machines.

2.3.6.2 Access Control Memorandum

The Access Control Memorandum establishes, to the extent appropriate, controls to permit authorized and prevent unauthorized access of the software.

The Code Team/Sponsor shall document access control measures in the Access Control Memorandum following the process in Appendix A. When specifying access control on a system-wide basis, document or provide reference to the Access Control Memorandum describing system specific controls.

2.3.6.3 Approved Users Memorandum

The Approved Users Memorandum identifies users for a particular code. Users may be identified by name, organization, group, readers of approved test and/or analysis plan, etc. The Approved Users Memorandum shall be included as part of Installation and Checkout Phase Documentation.

The Code Team/Sponsor shall document approved users in the Approved Users Memorandum following the process in Appendix A.

Note: User list may be changed without modification of the Software Installation and Checkout Form QAP 19-1-8 (Appendix I).

2.3.7 Maintenance Phase

This section provides the process for requesting, controlling and implementing changes to software configuration baselines. Changes to software production baselines shall be formally evaluated, approved or disapproved, and the change appropriately reflected in associated baseline documentation.

2.3.7.1 Production Software and/or Baseline Document Change Control

When necessary, the Code Team/Sponsor shall propose changes to the software baseline, following the process in Appendix A and using the Change Control Form, Form QAP 19-1-9, (Appendix J),

Major changes – include new requirements, new design, new models, and new implementation, require a new baseline (i.e., SQAP, RD, DD, VVP, ID, UM, VD) to be documented. In addition to revising every baseline document a change control form and the Installation and Checkout Form are used.

Minor changes – do not affect the requirements or design and can be documented with addenda (no more than three addenda's per baseline document) or page changes to the affected baseline document, in addition to the Change Control form and the Installation and Checkout Form.

Patch changes – can be used for very small fixes to the code usually one or two lines of source code or expanding a fields character length etc. Patch changes can be documented and tested with the Change Control Form and Installation & Checkout Form.

The SCM Coordinator shall:

- identify affected software configuration baselines.
- verify unique revision identifier.
- inform affected users of approved changes. **Note**: If an organization is listed as an approved user, the organization's manager will be notified.
- redline/update baseline list.
- maintain a copy of the Change Control Form and forward to the SNL Records Center.

The Code Team/Sponsor shall:

- Perform modifications to software and/or associated baseline documentation in accordance with the appropriate sub-sections of this procedure. The version of the revision(s) should reflect the nature and scope of the change (see Section 2.0).
- Ensure that all baseline component identifiers are consistent (see Section 2.0)
- If modifications require re-compilation of the software, perform regression testing as identified in the approved VVP. Document per the Installation and Checkout phase of Section 2.3.6. The degree of software validation shall be reasonable and commensurate with the nature and scope of the change.

Note: If the software was modified to correct a problem, Code Team/Sponsor shall ensure that the Software Problem Reporting (SPR) process (Section 2.3.7.3) has been initiated.

2.3.7.2 System Software and Hardware Change Control

2.3.7.2.1 Coding Documentation Standards

Any change to software must be accompanied by documentation describing the change, the date the change was made, and the name of the person responsible for implementing the change. This documentation should be clearly identified, and placed in the code in the vicinity of the change, as well as at the top of the code prior to the first executable line. The code reviewer shall determine if this documentation is clear and sufficient.

2.3.7.2.2 Significant System Software or Hardware Changes

The Code Team/Sponsor (single-user systems) or System Administrator (multi-user systems) shall propose significant system software or hardware changes following the process described in Appendix A, using the Change Control Form QAP 19-1-9 (Appendix J).

Examples of significant changes to system software or hardware:

- changes to the operating system such that the version or level identifier changes
- changes to the Central Processing Unit (CPU)
- database management system change

In general, changes are significant if they impact the results generated by production software or cause recompilation of production software.

The Code Team/Sponsor or System Administrator shall:

- perform the approved system modification to the system software and/or hardware.
- perform regression testing (after significant changes have been performed on the production computer and prior to the next use of the baseline software) on all affected production baseline software in accordance with Section 2.3.6, Installation and Checkout.

2.3.7.3 Software Problem Report (SPR)

Whenever a software problem is identified, the Code Team/Sponsor shall evaluate the problem to determine if it is indeed a problem (as opposed to user error). If it is a problem, the SPR process shall be followed.

The Code Team/Sponsor shall classify the problem as major if it could significantly impact previous uses of code or if it will require significant modification to the software; otherwise classify it as minor.

The Code Team/Sponsor shall complete Part I of the Software Problem Report and Evaluation Form, QAP 19-1-10, Software Problem Report Form, and forward it to the Responsible Manager for concurrence on classification (i.e., major, minor).

The SCM Coordinator shall assign an SPR number in Field 2 of the Software Problem Report and Evaluation Form, and notify Responsible Manager by sending a copy of the Software Problem Report. The SCM Coordinator shall also update/redline the Software Baseline List.

For *major problems*, the Responsible Manager shall identify affected users to be notified of the problem, and designate qualified personnel to identify and evaluate the impact of the software problem. The affected analyses shall be revised as necessary, and the evaluation and resolution of the software problem shall be documented in Part II of the Software Problem Report and Evaluation Form. For *minor problems,* this evaluation can be performed by the Code Team/Sponsor.

The responsible manager shall approve the evaluation and resolution by signing the form and forwarding it to the SCM Coordinator.

The SCM Coordinator shall retain copies of the form, transmit the original record to OCRWM and forward a copy to the SNL Records Center.

If necessary, the Code Team/Sponsor shall propose changes to correct the applicable baseline components per Section 2.3.7.1.

2.3.7.4 Configuration Management (Configuration Identification and Status Accounting)

This section provides the process for defining the configuration of software products, establishing software configuration baselines, and tracking the status of baseline changes. A software configuration baseline consists of the source code and baseline documents, providing objective evidence of technical adequacy. The process for preparation and approval of software baselines is described in Appendix A.

The SCM Coordinator shall maintain a Software Baseline List, and make it available upon request. The SCM Coordinator performs a completeness review to ensure compliance with the procedure, and to ensure that necessary components of configuration management are present.

The Software Baseline List shall contain:

- code name and version,
- code version date,
- Code Team/Sponsor name,
- code classification (see Appendix A),
- RD version,
- VVP version,
- DD version,
- ID version,
- UM version,
- VD version,
- list of approved users (may be listed by name, organization, group, or task, etc…)
- list of approved system software/hardware configurations,
- list of outstanding Software Problem Report (SPR) numbers (see Section 2.3.7.3), and
- status of approved changes that are in process.
- I&C date

The SCM coordinator shall redline the Software Baseline List when new or revised software products and/or documentation baselines are approved for use. A redlined list shall be maintained until a new baseline list is issued. The SCM coordinator shall periodically (at least once every calendar year), issue the baseline software list identifying all software with no approved users as candidates for retirement.

The Code Team/Sponsors shall review the Software Baseline List for accuracy and for codes that may be retired from production use. (Code retirement is addressed in Section 2.3.8). Code Team/Sponsors shall report any changes or inaccuracies to the SCM Coordinator.

2.3.8 Retirement Phase

To retire a code, the Code Team/Sponsor issues a memorandum to the SCM Coordinator requesting that the code be retired, and provide a reason for the retirement.

The SCM Coordinator marks the code as retired in the baseline software list.

The System Administrator and/or Code Team/Sponsor shall take action to prevent the use of the retired code. This could involve removal of the software from the computer or the changing of execution privileges.

2.4 Interim Use of Unqualified Software

With written permission granted in advance by the Responsible Manager relying on input from Software Coordinator, some software that is required to support various OSTI activities may need to be used prior to full qualification.

Software covered by this section is not to be used for any other purposes or any other milestone deliverables. This section describes the requirements and process methodologies that will permit the interim use and controls of unqualified software in products that are currently being developed to support the SNL OSTI program.

2.4.1 Code Team/Sponsor

Determine the need to use unqualified software based on the work scope, deliverable schedule, and complexity of confirmation once qualified. Prepare a Test Plan (TP) in accordance with QAP 20-1, Test Plans. The AP, in this case, shall outline how the unqualified software will be used, a schedule for qualification, and comparison confirmation methodologies, including acceptance criteria to be used to determine the extent of impact evaluations that may be applicable once the software is qualified.

2.4.2 OSTI Team Lead or Designee

Approve the Test Plan by signing the cover signature page. **NOTE**: This signature serves as the written permission.

2.4.3 SCM Coordinator

Establish and maintain an unqualified software list containing the code name and version, version date, System Configuration, Code Team/Sponsor, and Code Classification.

2.4.4 Code Team/Sponsor

a) Install the unqualified software in accordance with Section 2.3.6 Installation and Checkout Phase and submit an Implementation Document per Section 2.3.4 for the Implementation phase. Initiate a CAR in accordance with QAP 16-1 to track the use of the data generated with the unqualified software.
b) Continue work on the documentation and qualification aspects of the software in accordance with this procedure.
c) Once the software has been qualified and baselined in accordance with this procedure, compare the test cases run on the qualified version with each of the test cases run on the unqualified software versions that were used to generate data, develop data or output.

1) If the comparison indicates that no differences exist or that the differences can be justified, all previous data generated from that version of software are acceptable. Justification for the differences must be documented.
2) If differences exist that cannot be justified, all previous data generated must be re-run, using the qualified version of the software.
3) Once the software has been qualified and baselined and the impact reviews have been resolved, submit the record copy to the SCM Coordinator for inclusion in the software records package.

If the software will not be used in a production environment then retire the software per Section 2.3.8 of this procedure.

3.0 Records

The following QA records, which may be generated through implementation of this procedure, shall be prepared and submitted to the CRWM Records Center in accordance with QAP 17-1, Records. A copy shall be submitted to the SNL Records Center as individual records or included in a records package.

The following records are submitted by the Technical Data Coordinator in accordance with this and/or relevant data generating procedure(s):

QA Record	Preparer	Records Submitter
• Software Baseline List	SCM Coordinator	SCM Coordinator
• Primitive Baseline Review Memorandum	Code Team/Sponsor	SCM Coordinator
• Software QA Plan	Code Team/Sponsor	SCM Coordinator
• Requirements Document (RD)	Code Team/Sponsor	SCM Coordinator
• Verification and Validation Plan (VVP)	Code Team/Sponsor	SCM Coordinator
• Design Document (DD)	Code Team/Sponsor	SCM Coordinator
• User's Manual (UM)	Code Team/Sponsor	SCM Coordinator
• Implementation Document (ID) (the source code may be stored in a configuration management tool in lieu of records)	Code Team/Sponsor	SCM Coordinator
• Validation Document (VD)	Code Team/Sponsor	SCM Coordinator
• Software QA Plan Criteria Form (QAP 19-1-1)	Code Team/Sponsor	SCM Coordinator
• Requirement Document Criteria Form (QAP 19-1-2)	Code Team/Sponsor	SCM Coordinator
• Verification and Validation Plan Criteria Form (QAP 19-1-3)	Code Team/Sponsor	SCM Coordinator
• Design Document Criteria Form (QAP 19-1-4)	Code Team/Sponsor	SCM Coordinator
• Implementation Document Criteria Form (QAP 19-1-5)	Code Team/Sponsor	SCM Coordinator
• User Manual Criteria Form (QAP 19-1-6)	Code Team/Sponsor	SCM Coordinator

QA Record	Preparer	Records Submitter
• Validation Document Criteria Form (QAP 19-1-7)	Code Team/Sponsor	SCM Coordinator
• Software Installation and Checkout Form (QAP 19-1-8)	Code Team/Sponsor	SCM Coordinator
• Change Control Form (QAP 19-1-9)	Code Team/Sponsor	SCM Coordinator
• Software Problem Reporting and Evaluation Form (QAP 19-1-10)	Code Team/Sponsor	SCM Coordinator
• SPR Closure Memorandum	Responsible Manager	SCM Coordinator
• Access Control Memorandum	Code Team/Sponsor	SCM Coordinator
• Approved User Change Memorandum	Code Team/Sponsor	SCM Coordinator
• Code Retirement Request Memorandum	Code Team/Sponsor	SCM Coordinator

4.0 Appendices

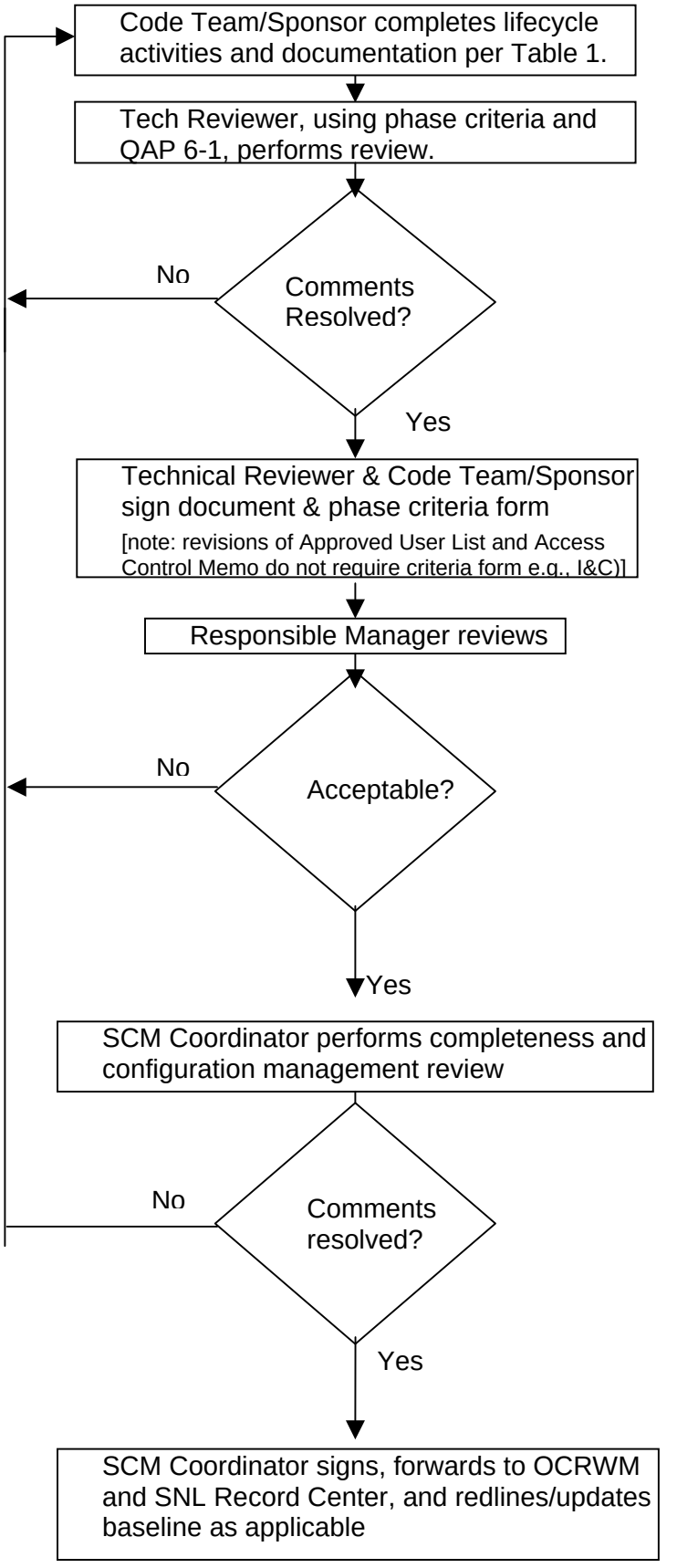
Appendix A: Software Life-Cycle Process Flow Chart
Appendix B: Form QAP 19-1-1, Software QA Plan Criteria Form
Appendix C: Form QAP 19-1-2, Requirements Document Criteria Form
Appendix D: Form QAP 19-1-3, Verification and Validation Plan Criteria Form
Appendix E: Form QAP 19-1-4, Design Document Criteria Form
Appendix F: Form QAP 19-1-5, Implementation Document Criteria Form
Appendix G: Form QAP 19-1-6, User's Manual Criteria Form
Appendix H: Form QAP 19-1-7, Validation Document Criteria Form
Appendix I: Form QAP 19-1-8, Software Installation and Checkout Form
Appendix J: Form QAP 19-1-9, Change Control Form
Appendix K: Form QAP 19-1-10, Software Problem Report (SPR)

5.0 References

QAP 16-1 Corrective Action
QAP 17-1 Records Management
LP-SI.11Q-BSC Software Management

Appendix A

OSTI Software Lifecycle Process Flow Chart



Appendix B

	OSTI Software QA Plan Criteria Form	Form Number: QAP 19-1-1 Page 1 of 1

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **OSTI Document #:** _____

Prior to sign-off of the SQA Plan, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the SQA Plan.

5. **Software Identification:** Are software name, version and scope identified answering why we are doing this and what problem will be solved?	☐ Yes
6. **Deviations:** If there are deviations from the Lifecycle required documentation, is the deviation adequately explained and is it appropriate?	☐ Yes ☐ N
7. **Documents:** Are the documents to be prepared, reviewed and maintained identified?	☐ Yes
8. **Organizations:** Are the organizations responsible for work and quality assurance identified with tasks (a schedule for qualification) and responsibilities?	☐
9. **Development Methods:** Are the standards, conventions, techniques, methods and procedures (QAPs, EIPs, etc.) identified for use in establishing and maintaining integrity of code data, embodied mathematical models, files, and processes?	☐ Yes
10. **Problem Reporting:** Is there a process for documenting and reporting software discrepancies, evaluating the impact of errors on previous calculations, and determining the appropriate corrective action(s)?	☐ Yes
11. **External Interfaces:** Are the interfaces with peripheral hardware, and other software identified?	☐ Yes
12. **Completeness:** Is the plan complete?	☐ Yes
13. **Verifiability:** Can meeting the plan be verified?	☐ Yes
14. **Consistency:** Is the plan consistent internally and with other software?	☐ Yes
15. **Technical Feasibility:** Is the plan technically feasible and can it result in a useable code?	☐ Yes

16.	**Code Team/Sponsor's** Name (*print*)	Signature	Date
17.	**Technical Reviewer's** Name (*print*)	Signature	Date
18.	**Responsible Manager's** Name (*print*)	Signature	Date
19.	**SCM Coordinator's** Name (*print*)	Signature	Date

Key for check boxes above:

Check **Yes** for each item reviewed and found acceptable.

Check **N/A** for items which are not applicable.

Appendix C

	OSTI Requirements Document Criteria Form	Form Number: QAP 19-1-2 Page 1 of 1

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **OSTI Document #:** _____

Prior to sign-off of the RD, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the RD.

5.	**Functionality:** Are the functions that the software is to perform adequately identified?	☐ Yes	
6.	**Performance:** Are time-related software operations issues, e.g., speed, recovery time, or response time identified, where applicable as based on the code functionality?	☐ Yes	☐ N/A
7.	**Design Constraints:** Are elements that will restrict design options identified?	☐ Yes	
8.	**Attributes (non-time-related):** Are the following identified, where applicable as based on the code functionality:		
	portability?	☐ Yes	☐ N/A
	acceptance criteria?	☐ Yes	
	maintainability?	☐ Yes	☐ N/A
9.	**External Interfaces:** The following interfaces identified, where applicable as based on the code functionality:		
	People?	☐ Yes	☐ N/A
	Hardware?	☐ Yes	☐ N/A
	Software?	☐ Yes	☐ N/A
10.	**Completeness:** Are the requirements complete?	☐ Yes	
11.	**Verifiability:** Is meeting the requirements be verified?	☐ Yes	
12.	**Consistency** Are requirements consistent with each other?	☐ Yes	
13.	**Technical Feasibility** Are the requirements technically feasible and can they result in a useable code?	☐ Yes	

14.	**Code Team/Sponsor's Name** (*print*)	Signature	Date
15.	**Technical Reviewer's Name** (*print*)	Signature	Date
16.	**Responsible Manager's Name** (*print*)	Signature	Date
17.	**SCM Coordinator's Name** (*print*)	Signature	Date

Key for check boxes above:

Check **Yes** for each item reviewed and found acceptable
Check **N/A** for items not applicable

Form QAP 19-1-2 Instructions

1 – 4. These fields are needed for configuration management. Please supply the software name and version for which the RD is being written. Provide the RD Document Version. Follow Version requirements listed in Section 2.0.

5. Functionality. Functional requirements define what the software product must accomplish

6. Performance. Clearly describe all required time performance issues.

7. Design Constraints. Clearly describe any functional requirements that will later restrict design options.

8. Attributes.

- Portability. Describe any requirements for using the code on more than one system
- Acceptance Criteria. Acceptable result for a given functional requirement. Often includes a quantification of acceptable error range per %. Acceptance criteria specifies the outputs and features required to demonstrate acceptable performance and provides a quantitative basis for each required output or feature to be evaluated.
- Maintainability. The structure and style of the requirements allow for necessary changes can be made.

9. External Interfaces. Describe any interfaces with users that will be functional requirements (GUI interfaces for example).

10. Completeness. Each requirement describes a result that must be achieved. All requirements together describe all functionality that the software product will provide.

11. Verifiability. Functional requirements must be implementable as source code.

12. Consistency. Individual requirements are not in conflict with each other.

13. Technical Feasibility. The requirements can be implemented under existing constraints.

Appendix D

	OSTI Verification and Validation Plan Criteria Form	Form Number: QAP 19-1-3 Page 1 of 1

1. Software Name: _____

2. Software Version: _____

3. Document Version: _____

4. OSTI Document #: _____

Prior to sign-off of the VVP, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the VVP.

5. Sufficient Test Cases ☐ Yes
Does the VVP identify sufficient test cases and acceptance criteria to ensure the final software and end product satisfies the requirements of the approved RD? (Check Yes if peer review is identified to fulfill validation requirements)

6. Adequacy of Test Cases ☐ Yes
Do the test cases demonstrate that the code adequately performs all intended functions and produces valid results for problems encompassing the range of permitted usage?

7. Operational Control ☐ Yes
If the software is used for operational control, do tests demonstrate required performance over the range of operation of the controlled function or process?

8. Unintended Functions ☐ Yes
Do the test cases show that the code does not perform any unintended function that either by itself or in combination with other functions can degrade the intended outcomes of the software?

9. Test Result Validation. (check one or more which are applicable as based on code functionality)
The test results will be compared to the following:

- hand calculations,	☐ Yes	☐ N/A
- manual inspection,	☐ Yes	☐ N/A
- calculations using comparable proven problems,	☐ Yes	☐ N/A
- empirical data or information from confirmed published data and correlation's and/or technical literature,	☐ Yes	☐ N/A
- other validated software of similar purpose,	☐ Yes	☐ N/A
- independent software of similar purpose.	☐ Yes	☐ N/A
- documented peer review will be performed.	☐ Yes	☐ N/A

Do the test cases describe how the code results will be validated? ☐ Yes

10. Does the VVP specify the following, where applicable as based on code functionality?

(a) specific tests and test sequence	☐ Yes	☐ N/A
(b) required ranges of input parameters	☐ Yes	☐ N/A
(c) identification of the stages at which testing is required	☐ Yes	☐ N/A
(d) criteria for establishing test cases	☐ Yes	☐ N/A
(e) requirements for testing logic branches	☐ Yes	☐ N/A
(f) requirements for hardware integration	☐ Yes	☐ N/A
(g) anticipated output values	☐ Yes	☐ N/A
(h) acceptance criteria	☐ Yes	☐ N/A

11. Installation and Regression Testing
Are test cases which are suitable for installation testing and regression testing identified in the set of verification and validation test cases? ☐ Yes

Code Team/Sponsor's Name (*print*)	Signature	Date
Technical Reviewer's Name (*print*)	Signature	Date
Responsible Manager's Name (*print*)	Signature	Date
SCM Coordinator's Name (*print*)	Signature	Date

Key for check boxes above:
Check **Yes** for each item reviewed and found acceptable
Check **N/A** for items not applicable, where applicable as based on code functionality

Appendix E

	OSTI Design Document Criteria Form	Form Number: QAP 19-1-4 Page 1 of 1

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **OSTI Document #:** _____

Prior to sign-off of the DD, all items shall be appropriately addressed by the code sponsor so that "**Yes**" may be checked.
Include this form as part of the DD.

Are the following appropriately defined and documented in the DD?

5. Major Software Components ☐ Yes
6. Technical description of the software with respect to: theoretical basis, embodied mathematical model, major control flow, control logic, and data structures ☐ Yes
7. Allowable or Prescribed Ranges for Inputs and Outputs ☐ Yes
8. Verifiability: Is the design verifiable through testing or other means? ☐ Yes
9. Consistency and Traceability: Is the design consistent within and traceable to the software's requirements? ☐ Yes
10. Technical Feasibility: Is the design technically feasible? ☐ Yes
11. Implementation: Is the design presented in sufficient detail to allow for implementation as computer software? ☐ Yes

12.	**Team/Sponsor** (*print*)	Signature	Date
13.	**Technical Reviewer** (*print*)	Signature	Date
14.	**Responsible Manager** (*print*)	Signature	Date
15.	**SCM Coordinator** (*print*)	Signature	Date

Key for check boxes above:

Check **Yes** for each item reviewed and found acceptable

Appendix F

	OSTI Implementation Document Criteria Form	**Form Number: QAP 19-1-5** **Page 1 of 1**

1. Software Name: _____

2. Software Version: _____

3. Document Version: _____

4. OSTI Document #: _____

Prior to sign-off of the ID, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the ID.

5. Source Code

- Is the source code provided? ☐ Yes ☐ N/A
- If applicable, is the change documentation in the source code clear and sufficient? ☐ Yes ☐ N/A

Note: If the source code is not controlled in a configuration management tool then a hardcopy of the source is required. (*Check "N/A" for commercially obtained software for which source code was not provided.*)

6. Coding Standards ☐ Yes ☐ N/A

Are the coding standards and conventions which were adhered to in the development of the software identified?

7. Coding Standards Implementation ☐ Yes ☐ N/A

Does the source code adhere to the coding standards and conventions defined in the ID?

8. Executable Generation ☐ Yes ☐ N/A

Was the executable generation process documented?

9. Implementation Requirements ☐ Yes ☐ N/A

Was the code implemented according to the requirements of the SRD and where applicable the DD?

10. Code Team/Sponsor's Name (*print*)	Signature	Date
11. Technical Reviewer's Name (*print*)	Signature	Date
12. Responsible Manager's Name (*print*)	Signature	Date
13. SCM Coordinator's Name (*print*)	Signature	Date

Key for check boxes above:

Check **Yes** for each item reviewed and found acceptable
Check **N/A** for items not applicable

Appendix G

	OSTI User's Manual Criteria Form	**Form Number: QAP 19-1-6** **Page 1 of 1**

Does the user's manual contain as appropriate:

1. **Software Name:** _____
2. **Software Version:** _____
3. **Document Version:** _____
4. **oSTI Document #:** _____

Prior to sign-off of the User's Manual, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the User's Manual.

5.	A statement(s) of functional requirements (consistent with those in the RD) and system limitations?	☐ Yes
6.	An explanation of the mathematical model and numerical models, where applicable as based on code functionality?	☐ Yes ☐ N/A
7.	Physical and mathematical assumptions, where applicable as based on code functionality?	☐ Yes ☐ N/A
8.	The capabilities and limitations inherent in the software?	☐ Yes
9.	Instructions that describe the user's interaction with the software?	☐ Yes
10.	The identification of input parameters, formats, and valid ranges?	☐ Yes
11.	Messages initiated as a result of improper input and how the user can respond?	☐ Yes
12.	The identification and description of output specifications and formats?	☐ Yes
13.	A description of any required training necessary to use the software?	☐ Yes
14.	The identification of components of the code that were not tested?	☐ Yes
15.		

16.	**Code Team/Sponsor** (*print*)	Signature	Date
17.	**Technical Reviewer** (*print*)	Signature	Date
18.	**Responsible Manager** (*print*)	Signature	Date
	SCM Coordinator (*print*)	Signature	Date

Key for check boxes above:

Check **Yes** for each item reviewed and found acceptable
Check **N/A** for items not applicable, where applicable as based on code functionality

Appendix H

	OSTI Validation Document Criteria Form	Form Number: QAP 19-1-7 Page 1 of 1

1. Software Name: _____

2. Software Version: _____

3. Document Version: _____

4. OSTI Document #: _____

Prior to sign-off of the VD, all items shall be appropriately addressed by the code sponsor so that "**Yes**" or "**N/A**" may be checked. Include this form as part of the VD.

5. Is the following information included, where applicable?

(a) computer program and version tested	☐ Yes	☐ N/A
(b) computer hardware and operating system used	☐ Yes	☐ N/A
(c) test equipment and calibrations	☐ Yes	☐ N/A
(d) date of test	☐ Yes	☐ N/A
(e) tester or data recorder	☐ Yes	☐ N/A
(f) simulation models used,	☐ Yes	☐ N/A
(g) test problem input and output files	☐ Yes	☐ N/A
(h) results and acceptability	☐ Yes	☐ N/A
(I) action taken in connection with any deviations noted	☐ Yes	☐ N/A

6. Test Result Validation

The test results were compared to the following (check one or more, where applicable as based on code functionality):

- hand calculations,	☐ Yes	☐ N/A
- manual inspection,	☐ Yes	☐ N/A
- calculations using comparable problems,	☐ Yes	☐ N/A
- empirical data & information from confirmed published data and correlations for technical literature,	☐ Yes	☐ N/A
- other validated software of similar purpose	☐ Yes	☐ N/A
- other independent software of similar purpose.	☐ Yes	☐ N/A

7. Test Documentation Acceptability ☐ Yes

Do the tests meet the acceptance criteria identified in the approved VVP?

8. Documentation Repeatability ☐ Yes

Are the tests documented in sufficient detail such that they can be repeated?

9. Master File Documentation ☐ Yes

Are the test case input and output files included in the Validation Document?

10. Understandability of Documentation ☐ Yes

Are the validation methods, test data, results, and conclusions documented in a form that can be understood by an independent, technically competent individual?

11.		
Code Team/Sponsor (*print*)	Signature	Date
12.		
Technical Reviewer (*print*)	Signature	Date
13.		
Responsible Manager (*print*)	Signature	Date
14.		
SCM Coordinator (*print*)	Signature	Date

Key for check boxes above:

Check **Yes** for each item reviewed and found acceptable
Check **N/A** for items not applicable

Form QAP 19-1-7 Instructions

The Code Team/Sponsor or designee (e.g. tester) shall execute the test cases and compare results to the acceptance criteria identified in the approved VVP. Any tests performed during the implementation phase which were not previously documented and reviewed should be formally documented, as appropriate, and the VVP revised to reflect the additional tests.

"Manual Inspection" in Item 6 refers to manual activities which do not involve numerical manipulations. These include visual inspection of table reformatting or plotting, and concurrence of qualitative acceptance criteria such as trends in results due to input parameter variations.

In order to allow for comparison of test results to other independent software of similar purpose, the following criteria must be met:

- comparison of test results to any of the four previously listed methods in Item 5 is impossible or impractical;
- the computer codes were independently developed. This means development by different individuals. This should include the use of different theoretical bases, use of different modeling strategies, or different mathematical models;
- validation of any theoretical basis or mathematical model which is not considered a conventional, generally accepted solution technique for that application must be performed via another method.

The tests should demonstrate the capability of the software to produce valid results for problems encompassing the range of permitted usage as defined by the User's Manual.

Appendix I

	OSTI Software Installation and Checkout Form	**Form Number: QAP 19-1-8** **Page 1 of 1**

General Information

1. Software Name: _____
2. Software Version: _____
3. OSTI Document #: _____
4. Code Classification: _____
   a. ID Document Revision identifier: _____
   b. VD Document/Revision identifier (to which the test cases are compared): _____

Executable Or Object Information

5. Executable or Object Name (*include path)*: _____
6. Executable Or Object Size (*Bytes*): _____
7. Executable Or Object Date: _____

Compilation Information

8. Hardware System: _____
9. Operating System: _____

Installation And Checkou rmati

10. Hardware Sys _____
    Ope Syste _____
12. Any PRs tstar ng? ☐ Yes ☐ No SPR No(s).: _____

ase In m ion

13. ector Pa MS Library: _____
14. P edure(s): _____
15. aries: _____
16. Input Files: _____
17. Output Files: _____
18. Test Evaluation:

Test results fully met specified acceptance criteria ☐ Yes

19. Access Control and Approved User Memo (included in I&C or referenced) are complete and sufficiently identify needed control measures? ☐ Yes

20. **Code Team/Sponsor (*print)***	Signature	Date
21. **Technical Reviewer (*print*)**	Signature	Date
22. **Responsible Manager (*print*)**	Signature	Date
23. **SCM Coordinator (*print*)**	Signature	Date

Appendix J

	OSTI Change Control Form **(Software/Hardware/Baseline Document)**	**Form Number:** **QAP 19-1-9** **Page 1 of 2**

1. Software Name: _____

2. Software Version Identifier: a) Current: _____ b) Proposed: _____

3. Software Classification: a) Current: _____ b) Proposed: _____

4. OSTI Document #: _____

5. Hardware/Software Platform: _____

6. Type of change: ☐ Major ☐ Minor ☐ Patch

7. Proposed Changes: (attach pages as needed)

Document Affected		Required Resolution			
Software QA Plan (SQAP)	☐ Yes	☐ No*	☐ Revision	☐ Page Change	☐ Addenda
Version No: _____		*Rationale _____			
New Version No: _____		_____			
Requirements Document (RD)	☐ Yes	☐ No*	☐ Revision	☐ Page Change	☐ Addenda
Version No: _____		*Rationale _____			
New Version No: _____		_____			
Verification and Validation Plan (VVP)	☐ Yes	☐ No*	☐ Revision	☐ Page Change	☐ Addenda
Version No: _____		*Rationale _____			
New Version No: _____		_____			
Design Document (DD)	☐ Yes	☐ No*	☐ Revision	☐ Page Change	☐ Addenda
Version No: _____		*Rationale _____			
New Version No: _____		_____			

Appendix J (continued)

	OSTI Change Control Form **(Software/Hardware/Baseline Document)**	**Form Number:** **QAP 19-1-9** **Page 2 of 2**

Validation Document (VD) ☐ Yes ☐ No* ☐ Revision ☐ Page Change ☐ Addenda

Version No: _____ *Rationale _____

New Version No: _____

Implementation Document (ID) ☐ Yes ☐ No* ☐ Revision ☐ Page Change ☐ Addenda

Version No: _____ *Rationale

New Version No: _____

User's Manual (UM) ☐ Yes ☐ No* ☐ Revision ☐ Page Change ☐ Addenda

Version No: _____ *Rationale _____

New Version No:

8. System Software/Hardware change Section:

9. _____ _____ _____

Code Team/Sponsor's Name *(print)* or Computer Administrator — *Signature* — Date

10. _____ _____ _____

Technical Reviewer's Name *(print)* — *Signature* — Date

11. ☐ Approval of Proposed Change ☐ Disapprove of Proposed Change *(provide justification, attach pages as needed)*

12. _____ _____ _____

Responsible Manager's Name *(print)* — *Signature* — Date

13. _____ _____ _____

SCM Coordinator's Name *(print)* — *Signature* — Date

Change Control Form Instructions (Form Number QAP 19-1-9)

This form is for proposal and approval of changes to production baseline software, changes to software documentation, and/or changes to system software and hardware. Changes to system software and hardware applies to systems which are used by more than one person for running production baseline software.

General Instructions

For each entry listed, additional pages may be attached as needed.

1. Software Name: Enter the name of the software. For proposed system changes, enter applicable information (e.g. operating system name, hardware, device, etc.).

2. Software Version Identifier: On (a) enter the current software version identifier as listed on the Software Baseline Inventory List. On (b) enter the proposed version identifier. For proposed system changes, enter current status information (e.g. operating system version)

3. Software Classification: On (a) enter the current classification (i.e. acquired, developed) and on (b) enter the proposed classification.

4. OSTI Document #: Assigned to Change Control Form, obtained by SCM Coordinator.

5. Hardware/Software Platform: Enter the hardware platform on which the software resides and any applicable system software (required for the execution and use of the production baseline software).

6. Type of change: Select a box from this line to indicate whether changes are major, minor, or patch.

7. Proposed changes: Use this section to describe in detail the changes each document will be undergoing. For each document, list the current document version number (as it appears on the Baseline Inventory List) and (if applicable) the new document version number.

   Implementation Document (ID): Check if this document is affected and how it will be updated. In general, all ID changes will be revisions, not addenda's. Describe what aspects of the coding will change.

   Requirements Document (RD): Describe any features that are being changed, added, or deleted. Describe if any requirements are moving from not tested to tested. Include a discussion of required test cases to demonstrate acceptable performance of new code features. Provide rational for regression testing if all existing test cases will not be rerun.

   Verification and Validation Plan (VVP): Describe test cases and acceptance criteria that are being changed, added, or deleted. Discuss how these test cases demonstrate that the code adequately performs all tested functions.

   Design Document (DD): Describe the extent of changes to the DD. Note how changes will be verifiable through testing or other means.

   Validation Document (VD): Describe if the VD will change to reflect changes to the VVP or will be updated for other reasons.

User Manuals (UM): Describe what user instructions will be changed, added, or deleted.

8. System Software/Hardware Change Section
   Describe proposed changes to system software and/or hardware. Describe expected impact, if any, to production baseline software which resides on the system. Describe how changes to system software and/or hardware will be tested. Discuss what regression testing of baseline software will be required or describe why no regression testing of production baseline software will be needed. If testing is needed, it must address the change to the system to verify that the change has been installed properly and works properly.

9. Code Sponsor or System Administrator Signature
   Code Sponsor signs for changes to baseline software.
   System Administrator signs for changes to system software/hardware.

10. Technical Reviewer Signature. Indicated concurrence with impact to baseline documentation. For system software / hardware indicates concurrence with evaluation of impact to production baseline codes.

11. Responsible Manager selects a box from this line to indicate whether changes are approved or disapproved. If changes are disapproved, explain why.

12. Responsible Manager signature. After signing form, RM forwards to SCM Coordinator.

13. SCM Coordinator Signature. SCM coordinator signs change control form or returns it to code sponsor or computer administrator for proper completion. After SCM coordinator signature, forwards form to affected users and QC.

Appendix K

	OSTI Software Problem Report (SPR) Form	Form Number: QAP 19-1-10 Page 1 of 2

PART I

SPR Tracking Information

SPR Number (obtain from SCM coordinator): _____

Software Name: _____

Software Version: _____

SPR Classification: Major ☐ OR Minor ☐ (For major SPRs, impact statement is [illegible] from each person designated by Responsible Manager)

Description of Error: (attach pages as need)

Analysis (attach pages as needed)

Title Author

Code Team/Sponsor Name(*print*) *Signature* Date

Responsible Manager Name(*print*) *Signature* Date

Appendix K (continued)

	OSTI Software Problem Report (SPR) Form **(Continuation Sheet: Impact Assessment)**	**Form Number: QAP 19-1-10** **Page 2 of 2**

PART II
SPR Tracking Information
SPR Number: _____
Software Name: _____
Software Version: _____
Analysis Title

_____

_____

Impact in Analysis (attach pages as needed)

_____

_____

_____

_____

Author/Approved User Name *(print)*	*Signature*	Date
Technical Reviewer Concurrence Name *(print)*	*Signature*	Date
Responsible Manager Name *(print)*	*Signature*	Date

Instructions:

SPR Tracking Information: Enter the SPR Number, Software Name, and Software Version on each page of the Software Problem Report form for tracking purposes. This item is completed by the Code Team/Sponsor/SCM Coordinator prior to issuing the request for the impact statement.

SPR Classification: Designate level of problem by checking major or minor identified by Responsible Manager

Major: requires a response from each user. Major problems are problems that may cause calculations to be re-run or may necessitate a change to all baseline documents.

Minor: SPRs designated as minor require a response from Code Team/Sponsor.

Analysis Title: Give the title of the analysis for which impact is being described . This item is completed by the Code Team/Sponsor/SCM Coordinator prior to issuing the request for the impact statement.

Impact in Analysis: Describe impact in software/analysis, which used the output, produced by the subject software version. If there was no impact, provide justification. If additional calculations are needed to assess the impact, attach these for technical review concurrence.

Author Signature: Signature of person making the impact assessment.

Technical Reviewer Concurrence: Signature of person who performed the technical review of the software or the analysis. This indicates concurrence with the statement of error and with the assessment of impact. Note: this person shall be independent of the Code Team/Sponsor.

Responsible Manager: Signature of responsible manager indicating that complete impact statements have been given for all impacted analysis . After signature, forward to the SCM Coordinator for submittal to RC.

Note: After all impact statements have been made and reviewed, the responsible manager shall issue a memo to the SCM coordinator stating that all impact statements have been submitted for SCM processing and that the SPR is closed. SCM Coordinator will forward memo to the RC after updating Software Baseline List.